Relation Extraction with Matrix Factorization and Universal Schemas

Sebastian Riedel
Department of Computer Science
University College London
s.riedel@ucl.ac.uk

Limin Yao, Andrew McCallum, Benjamin M. Marlin
Department of Computer Science
University of Massachusetts at Amherst
{lmyao,mccallum,marlin}@cs.umass.edu

Abstract

Traditional relation extraction predicts relations within some fixed and finite target schema. Machine learning approaches to this task require either manual annotation or, in the case of distant supervision, existing structured sources of the same schema. The need for existing datasets can be avoided by using a *universal schema*: the union of all involved schemas (surface form predicates as in OpenIE, and relations in the schemas of pre-existing databases). This schema has an almost unlimited set of relations (due to surface forms), and supports integration with existing structured data (through the relation types of existing databases). To populate a database of such schema we present matrix factorization models that learn latent feature vectors for entity tuples and relations. We show that such latent models achieve substantially higher accuracy than a traditional classification approach. More importantly, by operating simultaneously on relations observed in text and in pre-existing structured DBs such as Freebase, we are able to reason about unstructured and structured data in mutually-supporting ways. By doing so our approach outperforms state-of-the-art distant supervision.

1 Introduction

Most previous work in relation extraction uses a predefined, finite and fixed schema of relation types (such as *born-in* or *employed-by*). Usually some textual data is labeled according to this schema, and this labeling is then used in supervised training of an automated relation extractor, e.g. Culotta and Sorensen (2004). However, labeling textual relations is time-consuming and difficult, leading to significant recent interest in distantly-supervised learning. Here one aligns existing database records with the sentences in which these records have been "rendered"—effectively labeling the text—and from this labeling we can train a machine learning system as before (Craven and Kumlien, 1999; Mintz et al., 2009; Bunescu and Mooney, 2007; Riedel et al., 2010). However, this method relies on the availability of a large database that has the desired schema.

The need for pre-existing datasets can be avoided by using language itself as the source of the schema. This is the approach taken by OpenIE (Etzioni et al., 2008). Here surface patterns between mentions of concepts serve as relations. This approach requires no supervision and has tremendous flexibility, but lacks the ability to generalize. For example, OpenIE may find FERGUSON–*historian-at*–HARVARD but does not know FERGUSON–*is-a-professor-at*–HARVARD. OpenIE has traditionally relied on a large diversity of textual expressions to provide good coverage. But this diversity is not always available, and, in any case, the lack of generalization greatly inhibits the ability to support reasoning.

One way to gain generalization is to cluster textual surface forms that have similar meaning (Lin and Pantel, 2001; Pantel et al., 2007; Yates and Etzioni, 2009; Yao et al., 2011). While the clusters discovered by all these methods usually contain semantically related items, closer inspection invariably shows that they do not provide reliable implicature. For example, a typical representative cluster may include *historian-at*, *professor-at*, *scientist-at*, *worked-at*. Although these relation types are indeed semantically related, note that *scientist-at* does not necessarily imply *professor-at*, and *worked-at*

Proceedings of NAACL-HLT 2013, pages 74–84,
Atlanta, Georgia, 9–14 June 2013.

certainly does not imply *scientist-at*. In fact, we contend that any relational schema would inherently be brittle and ill-defined—having ambiguities, problematic boundary cases, and incompleteness.[1] For example, Freebase, in spite of its extensive effort towards high coverage, has no *critized* nor *scientist-at* relation.

In response to this problem, we present a new approach: implicature with *universal schemas*. Here we embrace the diversity and ambiguity of original inputs; we avoid forcing textual meaning into predefined boxes. This is accomplished by defining our schema to be the union of all source schemas: original input forms, e.g. variants of surface patterns similarly to OpenIE, as well as relations in the schemas of many available pre-existing structured databases. But then, unlike OpenIE, our focus lies on learning asymmetric implicature among relations. This allows us to probabilistically "fill in" inferred unobserved entity-entity relations in this union. For example, after observing FERGUSON–*historian-at*–HARVARD our system infers that FERGUSON–*professor-at*–HARVARD, but not vice versa.

At the heart of our approach is the hypothesis that we should concentrate on predicting source data—a relatively well defined task that can be evaluated and optimized—as opposed to modeling semantic equivalence, which we believe will always be illusive.

Note that by operating simultaneously on relations observed in text and in pre-existing structured databases such as Freebase, we are able to reason about unstructured and structured data in mutually-supporting ways. For example, we can predict surface pattern relations that effectively serve as additional features when predicting Freebase relations, hence improving generalization. Also notice that users of our system will not have to study and understand the complexities of a particular schema in order to issue queries; they can ask in whatever form naturally occurs to them, and our system will likely already have that relation in our universal schema.

Our technical approach is based on extensions to probabilistic models of matrix factorization and collaborative filtering (Collins et al., 2001; Koren, 2008; Rendle et al., 2009). We represent the probabilistic knowledge base as a matrix with entity-entity pairs in the rows and relations in the columns (see figure 1). The rows come from running cross-document entity resolution across pre-existing structured databases and textual corpora. The columns come from the union of surface forms and DB relations. We present a series of models that learn lower dimensional manifolds for tuples, relations and entities, and a set of weights that capture direct correlations between relations. Weights and lower dimensional representations act, through dot products, as the natural parameters of a single log-linear model to derive per-cell probabilities.

In experiments we show that our models can accurately predict surface patterns relationships which do not appear explicitly in text, and that learning latent representations of entities, tuples and relations substantially improves results over a traditional classifier approach. Moreover, we can improve accuracy by simultaneously operating on relations observed in the New York Times corpus and in Freebase. In particular, our model outperforms the current state-of-the-art distant supervision method (Surdeanu et al., 2012) by 10% points Mean Average Precision through joint implicature among surface patterns and Freebase relations.

2 Model

Before we present our approach in more detail, we briefly introduce some notation. We use $\mathcal{R}$ to denote the set of relations we seek to predict (such as *works-written* in Freebase, or the X–*historian-at*–Y pattern), and $\mathcal{T}$ to denote the set of input tuples. For simplicity we assume each relation to be binary, although our approach can be easily generalized to the n-ary case. Given a relation $r \in \mathcal{R}$ and a tuple $t \in \mathcal{T}$ the pair $\langle r, t \rangle$ is a *fact*, or relation instance. The input to our model is a set of observed facts $\mathcal{O}$, and the observed facts for a given tuple is denoted by $\mathcal{O}_t := \{\langle r, t \rangle \in \mathcal{O}\}$.

Our goal is a model that can estimate, for a given relation r (such as X–*historian-at*–Y) and a given tuple t (such as <FERGUSON,HARVARD>), the probability $p(y_{r,t} = 1)$ where $y_{r,t}$ is a binary random variable that is true iff t is in relation r. We

[1] At NAACL 2012 Lucy Vanderwende asked "Where do the relation types come from?" There was no satisfying answer. At the same meeting, and in line with Brachman (1983), Ed Hovy stated "We don't even know what is-a means."

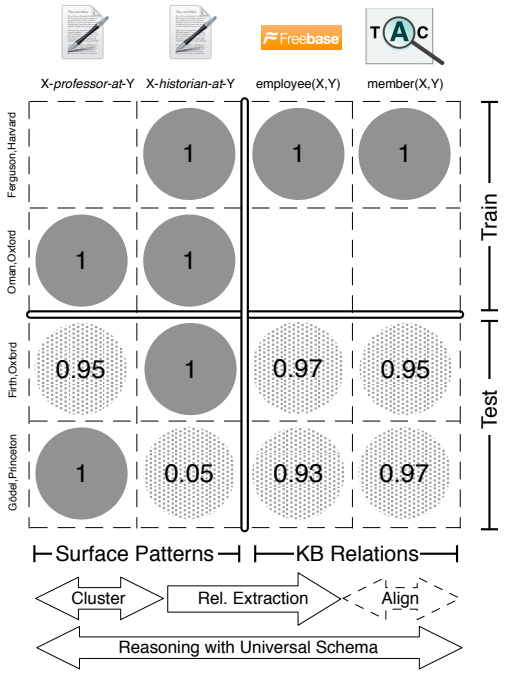

Figure 1: Filling up a database of universal schema. Dark circles are observed facts, shaded circles are inferred facts. Relation Extraction (RE) maps surface pattern relations (and other features) to structured relations. Surface form clustering models correlations between patterns, and can be fed into RE (Yao et al., 2011). Database alignment and integration models correlations between structured relations (not done in this work). Reasoning with the universal schema incorporates these tasks in a joint fashion.

introduce a series of exponential family models that estimate this probability using a *natural parameter* $\theta_{r,t}$ and the logistic function:

$$p\left(y_{r,t}=1|\theta_{r,t}\right) := \sigma\left(\theta_{r,t}\right) = \frac{1}{1+\exp\left(-\theta_{r,t}\right)}.$$

We will first describe our models through different definitions of the natural parameter $\theta_{r,t}$. In each case $\theta_{r,t}$ will be a function of r, t and a set of weights and/or latent feature vectors. In section 2.5 we will then show how these weights and vectors can be estimated based on the observed facts $\mathcal{O}$.

Notice that we can interpret $p\left(y_{r,t}=1\right)$ as the probability that a customer t likes product r. This analogy allows us to draw from a large body of work in collaborative filtering, such as work in probabilistic matrix factorization and implicit feedback.

2.1 Latent Feature Model

One way to define $\theta_{r,t}$ is through a latent feature model F. Here we measure compatibility between relation r and tuple t as dot product of two latent feature representations of size K^{F}: $\mathbf{a}_r$ for relation r, and $\mathbf{v}_t$ for tuple t. This gives:

$$\theta^{\mathrm{F}}_{r,t} := \sum_{k}^{K^{\mathrm{F}}} a_{r,k} v_{t,k}.$$

This corresponds to generalized PCA (Collins et al., 2001), a model were the matrix $\Theta = (\theta_{r,t})$ of natural parameters is defined as the low rank factorization $\mathbf{AV}$.

Notice that we intentionally omit any per-relation bias-terms. In section 4 we evaluate ranked answers to queries on a per-relation basis, and a per-relation bias term will have no effect on ranking facts of the same relation. Also consider that such latent feature models can capture asymmetry by assigning more peaked vectors to specific relations, and more uniform vectors to general relations.

2.2 Neighborhood Model

We can interpolate the confidence for a given tuple and relation based on the trueness of other similar relations for the same tuple. In collaborative filtering this is referred to as a *neighborhood-based* approach (Koren, 2008). In terms of our natural parameter, we implement a neighborhood model N via a set of *weights* $w_{r,r'}$, where each corresponds to a directed association strength between relations r and r'. For a given tuple t and relation r we then sum up the weights corresponding to all relations r' that have been observed for tuple t:

$$\theta^{\mathrm{N}}_{r,t} := \sum_{(r',t)\in\mathcal{O}\setminus\{(r,t)\}} w_{r,r'}.$$

Notice that the neighborhood model amounts to a collection of local log-linear classifiers, one for each relation r with feature functions $f_{r,r'}\left(t\right) = \mathbb{I}\left[r' \neq r \wedge (r',t) \in \mathcal{O}\right]$ and weights $\mathbf{w}_r$. This means that in contrast to model F, this model cannot harness any synergies between textual and pre-existing DB relations.

2.3 Entity Model

Relations have selectional preferences: they allow only certain types in their argument slots. While knowledge bases such as Freebase or DBPedia have extensive ontologies of types of entities, these are often not sufficiently fine to allow relations to discriminate (Yao et al., 2012b). Hence, instead of using a predetermined set of entity types, in our *entity model* E we learn a latent entity representation from data. More concretely, for each entity e we introduce a latent feature vector $\mathbf{t}_e$ of dimension K^{E}. In addition, for each relation r and argument slot i we introduce a feature vector $\mathbf{d}_i$ of the same dimension. For example, binary relations have feature representations $\mathbf{d}_1$ for argument 1, and $\mathbf{d}_2$ for argument 2. Measuring compatibility of an entity tuple and relation amounts to measuring, and summing up, compatibility between each argument slot representation and the corresponding entity representation. This leads to:

$$\theta^{\mathrm{E}}_{r,t} := \sum_{i=1}^{\mathrm{arity}(r)} \sum_{k}^{K^{\mathrm{E}}} d_{i,k} t_{t_i,k}.$$

Note that due to entity resolution, tuples may share entities, and hence parameters are shared across rows.

2.4 Combined Model

In practice all the above models can capture important aspects of the data. Hence we also use various combinations, such as:

$$\theta^{\mathrm{NFE}}_{r,t} := \theta^{N}_{r,t} + \theta^{\mathrm{F}}_{r,t} + \theta^{\mathrm{E}}_{r,t}.$$

2.5 Parameter Estimation

Our models are parametrized through weights and latent component vectors. We could estimate these parameters by maximizing the loglikelihood of the observed data akin to Collins et al. (2001). However, as we do not have access to negative facts, the model would simply learn to predict all facts to be true. In our initial attempt to overcome this issue we sampled a set of unobserved facts as designated negative facts, as is done in related distant supervision approaches. However, we found that (a) our results were sensitive to the choice of negative data and (b) runtime was increased substantially because of a large number of required negative facts.

In collaborative filtering positive-only data is also known as *implicit feedback*. This type of feedback arises, for example, when users buy but not rate items. One successful approach to learning with implicit feedback is based on the observation that the actual task is not necessarily one of prediction (here: to predict a number between 0 and 1) but one of (generally simpler) ranking: to give true "user-item" cells higher scores than false ones. *Bayesian Personalized Ranking* (BPR) uses a variant of this ranking: giving observed true facts higher scores than unobserved (true or false) facts (Rendle et al., 2009). This relaxed constraint is to be contrasted with the loglikelihood setting that essentially requires (randomly sampled) negative facts to score below a globally defined threshold.

2.5.1 Objective

We first create a dataset of *ranked pairs*: for each relation r and each observed fact $f^+ := \langle r, t^+ \rangle \in \mathcal{O}$ we choose all tuples t^- such that $f^- := \langle r, t^- \rangle \notin \mathcal{O}$—that is, tuples we have not observed to be in relation r. For each pair of facts f^+ and f^- we want $p(f^+) > p(f^-)$ and hence $\theta_{f^+} > \theta_{f^-}$. In BPR this is achieved by maximizing a sum terms of the form $\mathrm{Obj}_{f^+,f^-} := \log\left(\sigma\left(\theta_{f^+} - \theta_{f^-}\right)\right)$, one for each ranked pair:

$$\mathrm{Obj} := \sum_{\langle r,t^+\rangle \in \mathcal{O}} \sum_{\langle r,t^-\rangle \notin \mathcal{O}} \mathrm{Obj}_{\langle r,t^+\rangle,\langle r,t^-\rangle}. \quad (1)$$

Notice that this objective differs slightly from the one used by Rendle et al. (2009). Consider tuples as users and items as relations. We rank different users with respect to the same item, while BPR ranks items with respect to the same user. Also notice that the BPR objective is an approximation to the per-relation AUC (area under the ROC curve), and hence directly correlated to what we want to achieve: well-ranked tuples per relation.

Note that all parameters are regularized with quadratic penalty which we omit here for brevity.

2.5.2 Optimization

To maximize the objective[2] in equation 1 we follow Rendle et al. (2009) and employ Stochastic Gradient Descent (SGD). In particular, in each epoch

[2]This objective is non-convex for all models excluding the N model.

we sample $|\mathcal{O}|$ facts with replacement from $\mathcal{O}$. For each sampled fact $\langle r, t^+ \rangle$ we then sample a tuple $t^- \in \mathcal{T}$ such that $\langle r, t^- \rangle \notin \mathcal{O}$ is not an observed fact. This gives us $|\mathcal{O}|$ fact pairs $\langle f^+, f^- \rangle$, and for each pair we do an SGD update using the corresponding gradients of Obj_{f^+,f^-}. For the F model the gradients correspond to those presented by Rendle et al. (2009). The remaining gradients are easy to derive; we omit details for brevity.

3 Related Work

This work extends a previous workshop paper (Yao et al., 2012a) by introducing the neighborhood and entity model, by working with the BPR objective, and by more extensive experiments.

Relational Clustering There is a large body of work aiming to discover latent relations by clustering surface patterns (Hasegawa et al., 2004; Shinyama and Sekine, 2006; Kok and Domingos, 2008; Yao et al., 2011; Takamatsu et al., 2011), or by inducing synonymy relationships between patterns independently of the entities (Yates and Etzioni, 2009; Pantel et al., 2007; Lin and Pantel, 2001). Our approach has a fundamentally different objective: we are not (primarily) interested in clusters of patterns or their semantic representation, but in predicting patterns where they are not observed. Moreover, these related methods rely on a *symmetric* notion of synonymy in which clustered patterns are assumed to have the same meaning. Our approach rejects this assumption in favor of a model which learns that certain patterns, or combinations thereof, *entail* others in one direction, but not necessarily the other. This is similar in spirit to work on learning entailment rules (Szpektor et al., 2004; Zanzotto et al., 2006; Szpektor and Dagan, 2008). However, for us even entailment rules are just a by-product of our goal to improve prediction, and it is this goal we directly optimize for and evaluate.

Matrix Factorization Our approach is also related to work on factorizing YAGO to predict new links (Nickel et al., 2012). The primary differences are that we include surface patterns in our schema, use a ranking objective, and learn latent vectors for entities and tuples. Likewise, matrix factorization in various flavors has received significant attention in the lexical semantics community, from LSA to recent work on non-negative sparse embeddings (Murphy et al., 2012). In our problem columns correspond to relations, and rows correspond to entity tuples. By contrast, there columns are words, and rows are contextual features such as "words in a local window." Consequently, our objective is to *complete* the matrix, whereas their objective is to learn better latent embeddings of words (which by themselves again cannot capture any sense of asymmetry).

OpenIE Open IE (Etzioni et al., 2008) extracts facts mentioned in text, but does not predict potential facts not mentioned in text. Finding answers requires explicit mentions, and hence suffers from lower recall for not-so-frequently mentioned facts. Methods that learn rules between textual patterns in OpenIE aim at a similar goal as our proposed approach (Schoenmackers et al., 2008; Schoenmackers et al., 2010). However, their approach is substantially more complex, requires a categorization of entities into fine grained entity types, and needs inference in high tree-width Markov Networks. By contrast, our approach is based on a single unified model, requires no entity types, and for us inferring a fact amounts to not more than a few dot products. In addition, in our Universal Schema approach OpenIE surface patterns are just one kind of relations, and our aim is populate relations of all kinds. In the future we may even include relations between entities and continuous attributes (say, gene expression measurements).

Distant Supervision In Distant Supervision (DS) a set of facts from pre-existing structured sources is aligned with surface patterns mentioned in text (Bunescu and Mooney, 2007; Mintz et al., 2009; Riedel et al., 2010; Hoffmann et al., 2011; Surdeanu et al., 2012), and this alignment is then used to train a relation extractor. A core difference to our approach is the number of target relations: In DS it is the relatively small schema size of the knowledge base, while we also include surface patterns. This allows us to answer more expressive queries. Moreover, by learning from surface-pattern correlations, our latent models induce feature representations for patterns that do not appear in the DS training set. As we will see in section 4, this allows us to outperform state-of-the-art DS models.

Never-Ending Learning and Bootstrapping Our latent feature models are capable of never-ending learning (Carlson et al., 2010). That is, we can continue to train these models with incoming data, even if no structured annotation is available. In bootstrapping approaches the current model is used to predict new relations, and these hypothesized relations are used as new supervision targets (i.e. self-training). By contrast, our model only strengthens the correlations between incoming co-occurring observations. This has the advantage that wrong predictions are less likely be reinforced, hence reducing the risk of semantic drift.

4 Experiments

How accurately can we fill a database of Universal Schema, and does reasoning jointly across a universal schema help to improve over more isolated approaches? In the following we seek to answer this question empirically. To this end we train our models on observed facts in a newswire corpus and Freebase, and then manually evaluate ranked predictions: first for structured relations and then for surface form relations.

4.1 Data

Following previous work (Riedel et al., 2010), our documents are taken from the NYTimes corpus (Sandhaus, 2008). Articles after 2000 are used as training corpus, articles from 1990 to 1999 as test corpus. We also split Freebase facts 50/50 into train and test facts, and their corresponding tuples into train and test tuples. Then we align training tuples with the training corpus, and test tuples with the test corpus. This alignment relies on a preprocessing step that links NER mentions in text with entities in Freebase. In our case we use a simple string-match heuristic to find this linking. Now we align an entity tuple $\langle t_1, t_2 \rangle$ with a pair of mentions $\langle m_1, m_2 \rangle$ in the same sentence if m_1 is linked to t_1 and m_2 to t_2. Based on this alignment we filter out all relations for which we find fewer than 10 tuples with mentions in text.

The above alignment and filtering process reduces the total number of tuples related according to Freebase to 16k: approximately 8k tuples with facts mentioned in the training set, and approximately 8k such tuples for the test set. In addition we have a set of approximately 200k training tuples for which both arguments appear in the same sentence and both can be linked to Freebase entities, but for which no Freebase fact is recorded. This can either be because they are not related, or simply because Freebase does not contain the relationship yet. We also have about 200k such tuples in the test set. To simplify evaluation, we create a *subsampled test set* by randomly choosing 10k of the original test set tuples.

The above alignment allows us to determine, for each tuple t, the observed facts $\mathcal{O}_t$ as follows. To find the surface pattern facts $\mathcal{O}_t^{\text{PAT}}$ for the tuple $t = \langle t_1, t_2 \rangle$ we extract, for each mention $m = \langle m_1, m_2 \rangle$ of t, the *lexicalized dependency path* p between m_1 and m_2. Then we add $\langle p, t \rangle$ to $\mathcal{O}_t^{\text{PAT}}$. For example, we get "<-*subj*<-head->*obj*->" for "M1 heads M2." Filtering out patterns with fewer than 10 mentions in text yields approximately 4k patterns. For training tuples we add as Freebase facts $\mathcal{O}_t^{\text{FB}}$ all facts $\langle r, t \rangle$ that appear in Freebase, and for which r has not been filtered out beforehand. For the test set $\mathcal{O}_t^{\text{FB}}$ remains empty. The total set of observed facts $\mathcal{O}_t$ is $\mathcal{O}_t^{\text{FB}} \cup \mathcal{O}_t^{\text{PAT}}$, and their union over all tuples forms the set of observed facts $\mathcal{O}$.

4.2 Evaluation

For evaluation we use collections of relations: surface patterns in one experiment and Freebase relations in the other. In either case we compare the competing systems with respect to their ranked results for each relation in the collection. Given this ranking task, our evaluation is inspired by the TREC competitions and work in information retrieval (Manning et al., 2008). That is, we treat each relation as query and receive the top 1000 (*run depth*) entity pairs from each system. Then we pool the top 100 (*pool depth*) answers from each system and manually judge their relevance or "truth." This gives a set of relevant results that we can use to calculate recall and precision measures. In particular, we can use these annotations to measure an *average precision* across the precision-recall curve, and an aggregate *mean average precision* (MAP) across all relations. This metric has shown to be very robust and stable (Manning et al., 2008). In addition we also present a weighted version of MAP (weighted MAP) in which the average precision for each re-

Relation	#	MI09	YA11	SU12	N	F	NF	NFE
person/company	103	0.67	0.64	0.70	0.73	0.75	0.76	**0.79**
location/containedby	74	0.48	0.51	0.54	0.43	0.68	0.67	**0.69**
author/works_written	29	0.50	0.51	0.52	0.45	0.61	0.63	**0.69**
person/nationality	28	0.14	**0.40**	0.13	0.13	0.19	0.18	0.21
parent/child	19	0.14	0.25	0.62	0.46	0.76	**0.78**	0.76
person/place_of_death	19	0.79	0.79	0.86	**0.89**	0.83	0.85	0.86
person/place_of_birth	18	0.78	0.75	0.82	0.50	0.83	0.81	**0.89**
neighborhood/neighborhood_of	12	0.00	0.00	0.08	0.43	0.65	0.66	**0.72**
person/parents	7	0.24	0.27	0.58	0.56	0.53	**0.58**	0.39
company/founders	4	0.25	0.25	0.53	0.24	0.77	**0.80**	0.68
film/directed_by	4	0.06	0.15	0.25	0.09	0.26	0.26	**0.30**
sports_team/league	4	0.00	0.43	0.18	0.21	0.59	**0.70**	0.63
team/arena_stadium	3	0.00	0.06	0.06	0.03	0.08	**0.09**	0.08
team_owner/teams_owned	2	0.00	0.50	0.70	0.55	0.38	0.61	**0.75**
roadcast/area_served	2	*1.00*	0.50	*1.00*	0.58	0.58	0.83	*1.00*
structure/architect	2	0.00	0.00	*1.00*	0.27	*1.00*	*1.00*	*1.00*
composer/compositions	2	0.00	0.00	0.00	0.50	0.67	**0.83**	0.12
person/religion	1	0.00	*1.00*	*1.00*	0.50	*1.00*	*1.00*	*1.00*
film/produced_by	1	*1.00*	*1.00*	*1.00*	*1.00*	0.50	0.50	0.33
MAP		0.32	0.42	0.56	0.45	0.61	0.66	0.63
Weighted MAP		0.48	0.52	0.57	0.52	0.66	0.67	0.69

Table 1: Average and (weighted) Mean Average Precisions for Freebase relations based on pooled results. The # column shows the number of true facts in the pool. NFE is statistically different to all but NF and F according to the sign test. Bold faced are winners per relation, italics indicate ties.

lation is weighted by the relation's number of true facts.

Notice that we deviate from previous work in distant supervision that (a) combines the results from several relations in a single precision recall curve, and (b) uses held-out evaluation to measure how well the predictions match existing Freebase facts. This has several benefits. First, when aggregating across relations results are often dominated by a few very frequent relations, such as *containedby*, providing little information about how the models perform across the board. Second, evaluating with Freebase held-out data is biased. For example, we find that frequently mentioned entity pairs are more likely to have relations in Freebase. Systems that rank such tuples higher receives higher precision than those that do not have such bias, regardless of how correct their predictions are. Third, we can aggregate per-relation comparisons to establish statistical significance, for example via the sign test.

Also note that while we run our models on the complete training and test set, evaluation is restricted to the subsampled test set.

4.3 Predicting Freebase Relations

Table 1 shows our results for Freebase relations, omitting those for which none of the systems can find any relevant facts. Our first baseline is MI09, a distantly supervised classifier based on the work of Mintz et al. (2009). This classifier only learns from observed pattern-relation pairs in the training set (of which we only have about 8k). By contrast, our latent feature models can learn pattern-pattern correlations both on the unlabeled training and test set (comparable to bootstrapping). We hence also compare against YA11, a version of MI09 that uses preprocessed cluster features according to Yao et al. (2011). The third baseline is SU12, the state-of-the-art Multi-Instance Multi-Label system by Surdeanu et al. (2012).

The remaining systems are our neighborhood

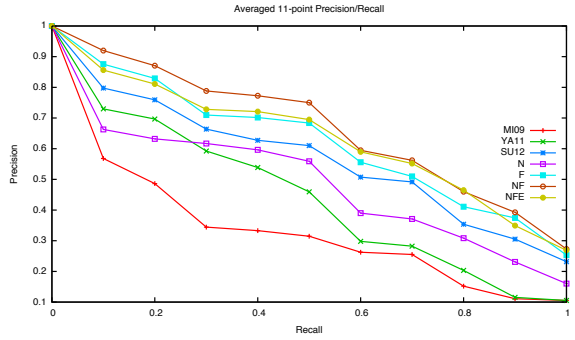

Figure 2: Averaged 11-point precision recall curve for Freebase relations in table 1.

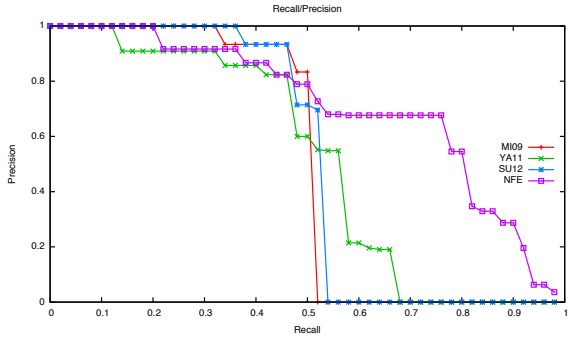

Figure 3: Precision and recall for *works_written*(X,Y).

model (N), the factorized model (F), their combination (NF) and the combined model with a latent entity representation (NFE). For all our models we use the same number of components when applicable ($K^{\mathrm{F}} = K^{\mathrm{E}} = 100$), 1000 epochs, and 0.01 as regularizer for component weights and 0.1 for neighborhood weights.

Table 1 shows that adding pattern cluster features (and hence incorporating more data) helps YA11 to improve over MI09. Likewise, we see that the factorized model F improves over N, again learning from unlabeled data. This improvement is bigger than the corresponding change between MI09 and YA11, possibly indicating that our latent representations are optimized directly towards improving prediction performance. The combination of N, F and E outperforms all other models in terms of weighted MAP, indicating the power of selectional preferences learned from data. Note that NFE is significantly different ($p \ll 0.05$ in sign test) to all but the NF and F models. In terms of MAP the NF model outperforms NFE, indicating that it does not do as well for frequent relations, but better for infrequent ones.

Figure 2 shows an averaged 11-point precision recall graph (Manning et al., 2008) for Freebase relations. We notice that our latent models outperform all remaining models across all recall levels, and that combining neighborhood and latent models is helpful. This finding is consistent with our MAP results. Figure 3 shows the recall-precision curve for the *works_written* relation with respect to our three baselines and the NFE model. Observe how precision drops for both MI09 and SU12 at about 50% recall. At this point the remaining unretrieved facts have patterns that have not been seen together with *works_written* in the training set. By using cluster features, YA11 can overcome this problem partly, but not as dramatically as NFE—a pattern we observe for many relations.

Relation	#	N	F	NF	NFE
visit	80	0.19	**0.68**	0.49	0.42
attend	69	**0.23**	0.10	0.07	0.10
base	61	0.46	**0.87**	0.81	0.68
head	38	0.47	0.67	**0.70**	0.68
scientist	36	0.25	**0.84**	0.79	0.73
support	18	0.16	0.29	0.32	**0.38**
adviser	11	0.19	0.15	0.19	**0.28**
criticize	9	0.09	0.60	**0.67**	0.64
praise	4	0.01	0.03	0.05	**0.10**
vote	3	0.18	0.18	**0.34**	0.34
MAP		0.22	0.44	0.44	0.43
Weighted MAP		0.28	0.56	0.50	0.46

Table 2: Average and (weighted) Mean Average Precisions for surface patterns.[2]

All our models are fast to train. The slowest model trains in just 45 minutes. By contrast, training the topic model in YA11 alone takes 4 hours. Training SU12 takes two hours (on less data). Also notice that our models not only learn to predict Freebase relations, but also approximately 4k surface pattern relations.

4.4 Predicting Surface Patterns

Table 2 presents a comparison of our models with respect to 10 surface pattern relations. These relations

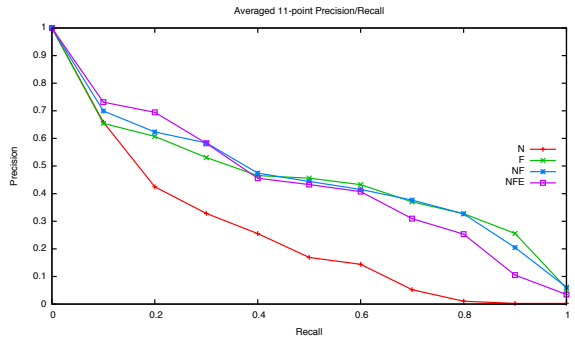

Figure 4: Averaged 11-point precision recall curve for surface pattern relations in table 2.

were chosen according to what we believe are interesting questions not currently captured in Freebase. We again see that learning a latent representation (F, NF and NFE) from additional data helps quite substantially over the N model. For in the weighted MAP metric we note that incorporating entity representations (in the NFE model) in fact hurts total performance.[3] One reason may be the fact that Freebase relations are typed—they require very specific types of entities as arguments. By contrast, for a surface pattern like "X visits Y" X could be a person or organization, and Y could be a location, organization or person. However, in terms of MAP score this time there is no obvious winner among the latent models. This is also confirmed by the averaged 11-point precision recall curve in figure 4.

Notice that we can accurately predict the X–*scientist-at*–Y surface pattern relation in table 2, as well as the more general *person/company* (employedBy) relation in table 1. This indicates that our models can capture asymmetry—a symmetric model would either over-predict X–*scientist-at*–Y or under-predict *person/company*.

5 Conclusion

We present relation extraction into universal schemas. Such schemas contain surface patterns as relations, as well as relations from structured sources. By predicting missing tuples for surface pattern relations we can populate a database without any labelled data, and answer questions not supported by the structured schema alone. By predicting missing tuples in the structured schema we can expand a knowledge base of fixed schema, and only require a set of existing facts from this schema. Crucially, by predicting and modeling both surface patterns and structured relations *simultaneously* we can improve performance. We show this experimentally by contrasting a series of the popular weakly supervised models to our collaborative filtering models that learn latent feature representations across surface patterns and structured relations. Moreover, our models are computationally efficient, requiring less time than comparable methods, while learning more relations.

Reasoning with universal schemas is not merely a tool for information extraction. It can also serve as a framework for various data integration tasks. For example, we could integrate facts from one schema (say, Freebase) into another (say, the TAC KBP schema) by adding both sets of relations to the set of surface patterns. Reasoning with this schema will mean populating each database with facts from the other, and would leverage information in surface patterns to improve integration. In future work we also plan to integrate universal entity types and attributes into the model.

The source code of our system, its output, and all data annotations are available at `http://www.riedelcastro.org/uschema`.

Acknowledgments

We thank the reviewers for very helpful comments. This work was supported in part by the Center for Intelligent Information Retrieval and the University of Massachusetts, in part by UPenn NSF medium IIS-0803847, in part by DARPA under agreement number FA8750-13-2-0020 and FA8750-09-C-0181, and in part by an award from Google. Any opinions, findings, and conclusion or recommendations expressed in this material are those of the authors and do not necessarily reflect the view of DARPA, AFRL, or the US government.

References

Ronald J. Brachman. 1983. What is-a is and isn t: An analysis of taxonomic links in semantic networks. *IEEE Computer*, 16(10):30–36.

[3] Due to the small set of relations only N is significantly different to F, NF and NFE ($p \ll 0.05$ in sign test).

Razvan C. Bunescu and Raymond J. Mooney. 2007. Learning to extract relations from the web using minimal supervision. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL '07)*.

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka, and Tom M. Mitchell. 2010. Toward an architecture for never-ending language learning. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI '10)*.

Michael Collins, Sanjoy Dasgupta, and Robert E. Schapire. 2001. A generalization of principal component analysis to the exponential family. In *Proceedings of NIPS*.

M. Craven and J. Kumlien. 1999. Constructing biological knowledge-bases by extracting information from text sources. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, pages 77–86, Germany.

Aron Culotta and Jeffery Sorensen. 2004. Dependency tree kernels for relation extraction. In *Proceedings of ACL*, Barcelona, Spain.

Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S. Weld. 2008. Open information extraction from the web. *Commun. ACM*, 51(12):68–74.

T. Hasegawa, S. Sekine, and R. Grishman. 2004. Discovering Relations among Named Entities from Large Corpora. *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL '04)*, pages 415–422.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S. Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of ACL*.

Stanley Kok and Pedro Domingos. 2008. Extracting Semantic Networks from Text Via Relational Clustering. In *ECML*.

Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, pages 426–434, New York, NY, USA. ACM.

Dekang Lin and Patrick Pantel. 2001. DIRT - discovery of inference rules from text. In *Knowledge Discovery and Data Mining*, pages 323–328.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, UK.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP (ACL '09)*, pages 1003–1011. Association for Computational Linguistics.

Brian Murphy, Partha Pratim Talukdar, and Tom Mitchell. 2012. Learning effective and interpretable semantic models using non-negative sparse embedding. In *COLING*, pages 1933–1950.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2012. Factorizing yago: scalable machine learning for linked data. In *Proceedings of the 21st international conference on World Wide Web*, WWW '12, pages 271–280, New York, NY, USA. ACM.

Patrick Pantel, Rahul Bhagat, Bonaventura Coppola, Timothy Chklovski, and Eduard Hovy. 2007. ISP: Learning Inferential Selectional Preferences. In *Proceedings of NAACL HLT*.

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 452–461, Arlington, Virginia, United States. AUAI Press.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD '10)*.

Evan Sandhaus, 2008. *The New York Times Annotated Corpus*. Linguistic Data Consortium, Philadelphia.

Stefan Schoenmackers, Oren Etzioni, and Daniel S. Weld. 2008. Scaling textual inference to the web. In *EMNLP '08: Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 79–88, Morristown, NJ, USA. Association for Computational Linguistics.

Stefan Schoenmackers, Oren Etzioni, Daniel S. Weld, and Jesse Davis. 2010. Learning first-order horn clauses from web text. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 1088–1098, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yusuke Shinyama and Satoshi Sekine. 2006. Preemptive information extraction using unrestricted relation discovery. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, HLT-NAACL '06, pages 304–311, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D. Manning. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings*

of the Conference on Empirical methods in natural language processing (EMNLP '12), pages 455–465.

Idan Szpektor and Ido Dagan. 2008. Learning entailment rules for unary templates. In *Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1*, COLING '08, pages 849–856, Stroudsburg, PA, USA. Association for Computational Linguistics.

Idan Szpektor, Hristo Tanev, Ido Dagan, and Bonaventura Coppola. 2004. Scaling web-based acquisition of entailment relations. In *Proceedings of EMNLP*.

Shingo Takamatsu, Issei Sato, and Hiroshi Nakagawa. 2011. Probabilistic matrix factorization leveraging contexts for unsupervised relation discovery. In *Proceedings of PAKDD*.

Limin Yao, Aria Haghighi, Sebastian Riedel, and Andrew McCallum. 2011. Structured relation discovery using generative models. In *Proceedings of the Conference on Empirical methods in natural language processing (EMNLP '11)*, July.

Limin Yao, Sebastian Riedel, and Andrew McCallum. 2012a. Probabilistic databases of universal schema. In *Proceedings of the AKBC-WEKEX Workshop at NAACL 2012*, June.

Limin Yao, Sebastian Riedel, and Andrew McCallum. 2012b. Unsupervised relation discovery with sense disambiguation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL '12)*, July.

Alexander Yates and Oren Etzioni. 2009. Unsupervised methods for determining object and relation synonyms on the web. *Journal of Artificial Intelligence Research*, 34:255–296.

Fabio Massimo Zanzotto, Marco Pennacchiotti, and Maria Teresa Pazienza. 2006. Discovering asymmetric entailment relations between verbs using selectional preferences. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics (ACL '06)*.